

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

**CORRELATIVE ASSESSMENT BETWEEN
SCANNED AND ORIGINAL DIGITAL IMAGES**

Inventor(s):

Mustafa Kesal

M. Kivanc Mihcak

Gary Starkweather

Ramarathnam Venkatesan

ATTORNEY'S DOCKET NO. MS1-1663US

CLIENT'S DOCKET NO. 305308.01

CORRELATIVE ASSESSMENT BETWEEN SCANNED AND ORIGINAL DIGITAL IMAGES

TECHNICAL FIELD

[0001] The described subject matter relates to electronic computing, and more particularly to a correlative assessment between scanned and original digital images.

BACKGROUND

[0002] Images such as photographs, pictures, logos, etc. may be created using digital rendering techniques, or using analog techniques which may be digitized. Digital images may be stored in a suitable storage medium such as, *e.g.*, a hard disk drive, CD-ROM, etc., and may be transmitted across digital communication networks for viewing and/or printing at remote locations.

[0003] In certain applications, *e.g.*, security and counterfeit detection, it may be desirable to distinguish between a printed copy of an original digital image and printed copy of a scanned copy of the original digital image or any other printed forgery copy of the original digital image.

SUMMARY

[0004] Implementations described and claimed herein address these and other issues by providing statistical approaches to distinguishing between a printed copy of an original digital image and printed copy of a scanned copy of the original digital image or any other printed forgery copy of the original digital

image. The approaches may be used independently or in conjunction with one another. The statistical approaches may be implemented as instructions on one or more computer-readable media which, when executed by a computing device, configure the computing device as a special-purpose machine that distinguishes between a printed copy of an original digital image and printed copy of a scanned copy of the original digital image or any other printed forgery copy of the original digital image.

[0005] In one exemplary implementation, a method is provided. The method comprises determining a correlation value between the scanned image and an original digital image, wherein the scanned image is included in a digital file, and generating a signal indicating whether the correlation value exceeds a threshold.

[0006] In another exemplary implementation, a computer-readable medium having computer-executable instructions is provided. The instructions, when executed, direct a computer to compare properties of an original digital image to properties of a scanned image of the original digital image, and generate a signal if a correlation value between properties of the original digital image and properties of the scanned image exceeds a threshold.

[0007] In another exemplary implementation, a computer program product comprising logic instructions executable on a processor is provided. The logic instructions comprise a scaling module that scales at least one of a first image file

and a second image file such that the files are of the same dimensions, and a correlation module that determines a correlation value between the first image file and the second image file, and generates a signal indicating whether the correlation value exceeds a threshold.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] Fig. 1 is a schematic illustration of an exemplary computing device that can be utilized to implement one or more described embodiments;

[0009] Fig. 2 is a schematic illustration of an exemplary software architecture for performing a statistical analysis of scanned images;

[0010] Fig. 3 is a flowchart illustrating operations in an exemplary procedure for performing a statistical analysis of scanned images;

[0011] Figs. 4a-4b are matrices illustrating an exemplary original digital image and a scanned image of a printed copy of the original digital image;

[0012] Fig. 5 is a flowchart illustrating operations in an exemplary process for scaling a raw, scanned image;

[0013] Fig. 6 is a flowchart illustrating exemplary operations for computing the variance deviation between a first digital image matrix and a second digital image matrix; and

[0014] Fig. 7 is a flowchart illustrating exemplary operations for calculating a calibration correlation value.

DETAILED DESCRIPTION

[0015] Described herein are exemplary methods for distinguishing between a printed copy of an original digital image and printed copy of a scanned copy of the original digital image or any other printed forgery copy of the original digital image. The methods described herein may be embodied as logic instructions on a computer-readable medium. When executed on a processor, the logic instructions cause a general purpose computing device to be programmed as a special-purpose machine that implements the described methods.

Exemplary Computing System

[0016] Fig. 1 is a schematic illustration of an exemplary computing device 130 on which logic instructions embodying the methods described herein may be executed.

[0017] Computing device 130 includes one or more processors or processing units 132, a system memory 134, and a bus 136 that couples various system components including the system memory 134 to processors 132. The bus 136 represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures. The system memory 134 includes read only memory (ROM) 138 and random access

memory (RAM) 140. A basic input/output system (BIOS) 142, containing the basic routines that help to transfer information between elements within computing device 130, such as during start-up, is stored in ROM 138.

[0018] Computing device 130 further includes a hard disk drive 144 for reading from and writing to a hard disk (not shown), a magnetic disk drive 146 for reading from and writing to a removable magnetic disk 148, and an optical disk drive 150 for reading from or writing to a removable optical disk 152 such as a CD ROM or other optical media. The hard disk drive 144, magnetic disk drive 146, and optical disk drive 150 are connected to the bus 136 by an SCSI interface 154 or some other appropriate interface. The drives and their associated computer-readable media provide nonvolatile storage of computer-readable instructions, data structures, program modules and other data for computing device 130. Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 148 and a removable optical disk 152, it should be appreciated by those skilled in the art that other types of computer-readable media which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, random access memories (RAMs), read only memories (ROMs), and the like, may also be used in the exemplary operating environment.

[0019] A number of program modules may be stored on the hard disk 144, magnetic disk 148, optical disk 152, ROM 138, or RAM 140, including an operating system 158, one or more application programs 160, other program

modules 162, and program data 164. A user may enter commands and information into computing device 130 through input devices such as a keyboard 166 and a pointing device 168. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are connected to the processing unit 132 through an interface 170 that is coupled to the bus 136. A monitor 172 or other type of display device is also connected to the bus 136 via an interface, such as a video adapter 174. In addition to the monitor, personal computers typically include other peripheral output devices (not shown) such as speakers and printers.

[0020] Computing device 130 commonly operates in a networked environment using logical connections to one or more remote computers, such as a remote computer 176. The remote computer 176 may be another personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to computing device 130, although only a memory storage device 178 has been illustrated in Fig. 1. The logical connections depicted in Fig. 1 include a local area network (LAN) 180 and a wide area network (WAN) 182. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets, and the Internet.

[0021] When used in a LAN networking environment, computing device 130 is connected to the local network 180 through a network interface or

adapter 184. When used in a WAN networking environment, computing device 130 typically includes a modem 186 or other means for establishing communications over the wide area network 182, such as the Internet. The modem 186, which may be internal or external, is connected to the bus 136 via a serial port interface 156. In a networked environment, program modules depicted relative to the computing device 130, or portions thereof, may be stored in the remote memory storage device. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

[0022] Generally, the data processors of computing device 130 are programmed by means of instructions stored at different times in the various computer-readable storage media of the computer. Programs and operating systems are typically distributed, for example, on floppy disks or CD-ROMs. From there, they are installed or loaded into the secondary memory of a computer. At execution, they are loaded at least partially into the computer's primary electronic memory. The invention described herein includes these and other various types of computer-readable storage media when such media contain instructions or programs for implementing the steps described below in conjunction with a microprocessor or other data processor. The invention also includes the computer itself when programmed according to the methods and techniques described below.

Exemplary Software Architecture

[0023] **Fig. 2** is a block diagram that shows further exemplary aspects of system memory 134 of Fig. 1, including application programs 160 and program data 164 for performing a statistical analysis of scanned images. In this implementation, application programs 160 include, for example an optional preprocessing module 202, scaling module 204, correlation module 206, and an optional calibration module 208. Program data 164 includes original image(s) 212, scaled image(s) 214, correlation values 216, and other data 218. Aspects of these computer-program modules and their operations are now described in detail in reference to exemplary images of Fig. 3 through Fig. 6.

Exemplary Operations

[0024] **Fig. 3** is a flowchart illustrating operations 300 in an exemplary correlation module 206 that performs a correlation analysis of scanned images. The operations 300 of correlation module 206 may be implemented as instructions on a computer-readable medium communicatively connected to a computing device such as, *e.g.*, computing device 130. Referring to Fig. 3, at operation 310 a scanned image is received at the computing device 130. In an exemplary implementation the scanned image may be received from an external digital scanning device. Digital scanning devices are known in the art. Broadly, digital

scanners scan a printed copy of an image and generate as output a digital file comprising the scanned image.

[0025] The digital file may be formatted in accordance with one or more digital file formats (*e.g.*, JPEG, GIF, PNG, BMP, etc.) and the scanned image may be in color, grayscale, or black-and-white (*i.e.*, binary) format. One skilled in the art will recognize that digital image files may be stored as a matrix, wherein each entry in the matrix includes a value (or an array of values). Each entry in the matrix corresponds to an image pixel, and each value corresponds in the entry represents a parameter of the pixel. For example, a black-and-white format image of size $(m \times n)$ may be stored as a matrix of $(m \times n)$ binary values. Similarly, a grayscale image of size $(m \times n)$ may be stored as a matrix of $(m \times n)$ integer values ranging between a lower bound of the grayscale (*e.g.*, 0) and an upper bound of the grayscale (*e.g.*, 255). A color image of size $(m \times n)$ may be stored as a matrix of $(m \times n)$ arrays. Each array may comprise a numeric value indicating a color and one or more other entries indicating one or more pixel properties (*e.g.*, a standard digital color image consists of pixels, where each pixel is represented by three numeric values; namely red, green and blue components, each which are integers in $\{0,1,2,\dots,254,255\}$).

[0026] **Figs. 4a-4b** are matrices illustrating an exemplary original digital image (Fig. 4a) and a scanned image (Fig. 4b) of a printed copy of the original digital image. The images illustrated in Figs. 4a-4b are in grayscale format,

wherein each element of the ($m \times n$) matrix comprises an integer entry between zero and 255, corresponding to a position on the grayscale. Neither the particular format nor the size of the images in Fig. 4a and Fig. 4b are critical.

[0027] In exemplary embodiments the scanned image may be received directly from a scanner, *e.g.*, over a suitable communication link. In alternate embodiments the scanned image may be received from an optional pre-processing module 202 that compensates for rotation and translation errors introduced by the scanning procedure. An exemplary pre-processing module is disclosed in a co-pending and commonly assigned U.S. patent application entitled ESTIMATING ORIENTATION ANGLE AND TRANSLATION VALUES OF SCANNED IMAGES, assigned Serial No. __/__,__, the entire disclosure of which is incorporated by reference. The pre-processing module 202 may be implemented as instructions executable in the processing unit 132 of computing device 130, or may be implemented in a processor associated with the scanner.

[0028] In addition to pre-processing, the scanned image may need to be scaled such that its dimensions correspond to the dimensions of the original digital image. Typically, the printing and scanning process result in the scanned image being significantly larger than the original digital image. By way of illustration, assume the original digital image of Fig. 4a is dimensioned as an ($m \times n$) matrix, which is printed to an image of size ($k \times l$) inches. The printed image is scanned at a resolution of P dots per inch to generate a raw scanned image that has

dimensions of $(k \cdot P \times l \cdot P)$. Hence, the raw scanned image may be reduced by a factor of $(k \cdot P/m)$ in the x dimension and $(l \cdot P/n)$ in the y dimension to reduce the scanned image to have dimensions of $(m \times n)$.

[0029] In an exemplary implementation the scaling process may be performed by computing an average value of the parameters in a region corresponding to a single pixel of the original image. **Fig. 5** is a flowchart illustrating operations 500 implemented in an exemplary scaling module 204 for scaling a raw, scanned image. At operation 510, estimate(s) of the original image(s) 212 may be retrieved from the raw, scanned image. This process is called the scaling of the raw scanned image. The scaling process begins with dividing the raw scanned image into $(m \times n)$ blocks, each having dimensions $(k \cdot P/m)$ in the x dimension and $(l \cdot P/n)$ in the y dimension. It is assumed that the raw scanned image is binary, *i.e.*, it consists of ones and zeros. If the scanner output consists of values in another range (*e.g.*, a typical scanned output may consist of integer values in $\{0,1,\dots,254,255\}$), then a suitable thresholding operation is initially performed on the scanner output to convert it to a binary image. Those, who are skilled in the art, would realize that this thresholding process may be application- and/or input-dependent (*e.g.*, the threshold value may be a constant value of 127 for the abovementioned example, or the threshold value may vary depending on the varying contrast generated by scanner to the scanner output). For each block of the raw scanned image, the following operation is performed: The empirical mean

of the pixel value parameters of the raw scanned image in that block is computed, then scaled by 255 and rounded to the nearest integer. This yields the average pixel value parameter of the estimate of the original image that corresponds to that block (operation 515).

[0030] If, at operation 520, the file format is not binary (i.e., the file format is either grayscale or color), then the average pixel value parameter for the block (found as a result of operation 515) may be assigned to the corresponding pixel in the scaled image (operation 530). Applying the notation $R(x,y)$ to refer to the (x,y) entries in the matrix corresponding to the raw, scanned image and $S(x,y)$ to refer to the entries in the matrix corresponding to the scanned image, the average value of the $R(x,y)$ pixels in a block are assigned to the corresponding $S(x,y)$ pixel. By contrast, if at operation 520, the pixel parameters are in a binary format, then a threshold value is applied to determine whether the corresponding $S(x,y)$ pixel should be assigned a binary zero or a binary one. For example, if the average value of the $R(x,y)$ pixels in a block exceed a threshold of 127, then the corresponding $S(x,y)$ pixel should be assigned a binary one. By contrast, if the average value of the $R(x,y)$ pixels in a block is less than a threshold of 127, then the corresponding $S(x,y)$ pixel should be assigned a binary zero.

[0031] The operations 515 through 530 are repeated for each of the blocks in the raw, scanned image, resulting in a matrix S having dimensions of $(m \times n)$. The matrix S may be stored in memory 134 as a scaled image 214. In subsequent

operations, the scaled image 214 may be compared directly to the original digital image matrix. In an alternate implementation, scaling module 204 may scale both images such that the scaled images have the same dimension.

[0032] Referring again to Fig. 3, at operation 315 a correlation value between the scanned image received in operation 310 (or the scaled version of the image) and the corresponding original digital image is determined. The corresponding original digital image may be stored locally, *e.g.*, as other data 226 in the memory 134 or, *e.g.*, on disk memory 152 of computing device 130. Alternatively, the corresponding original digital image may be stored remotely, *e.g.*, in a remote memory location communicatively connected to the computing device, and may be transmitted to the computing device 130 over a suitable communication link.

[0033] In one exemplary implementation, determining a correlation value comprises performing a pixel-by-pixel comparison of a property of the original image with a corresponding property of the scanned image. One technique comprises computing the variance σ^2 (or the standard deviation σ) between values representing the property in the original digital image and corresponding values in the scanned image. Applying the notation $D(x, y)$ to refer to the entries in the matrix corresponding to the original digital image and $S(x, y)$ to refer to the entries in the matrix corresponding to the scanned image, the variance σ^2 may be determined using the following equation:

$$\sigma^2 = \sum_{x=1}^m \sum_{y=1}^n (D(x,y) - S(x,y))^2 \quad \text{Equation (1)}$$

Here, we assume that the empirical means of both the original digital image D and the scanned image S is zero. Equivalently, we assume that prior to computing the variance via Equation (1), we subtract the empirical mean of D (resp. the empirical mean of S) from each D(x,y) (resp. from each S(x,y)) to make them zero-mean.

[0034] **Fig. 6** is a flowchart illustrating exemplary operations 600 for implementing Equation (1) to compute the variance between a first digital image matrix and a second digital image matrix. At operation 610, counters x and y are set to 1, and the value of σ^2 is set to zero. At operation 615, S(x,y) is subtracted from D(x,y) and the difference is squared and assigned to σ^2 . At operation 620 y is incremented and if, at operation 625, y is less than n, then control passes back to operation 615 and S(x,y) is subtracted from D(x,y) and the difference squared and added to σ^2 .

[0035] If, at operation 625, y is greater than or equal to n, then control passes to operation 630 and x is incremented. If at operation 635, x is less than m, then control passes back to operation 615 and S(x,y) is subtracted from D(x,y) and the difference squared and added to σ^2 . By contrast, if, at operation 635, x is greater than or equal to m, then control passes to 640, and the operation ends. The nested loop operations 615-635 compute the variance σ^2 between the matrix S(x,y) that represents the scanned image and the matrix D(x,y) that represents the original digital image. The variance σ^2 may be stored as a correlation value 216 in

memory 134. One skilled in the art will recognize that the standard deviation or a higher-order difference may be used in place of the standard deviation.

[0036] In another exemplary implementation, determining a correlation value comprises computing a cross-product between values representing the property in the original digital image and corresponding values in the scanned image. Again, applying the notation $D(x,y)$ to refer to the entries in the matrix corresponding to the original digital image and $S(x,y)$ to refer to the entries in the matrix corresponding to the scanned image, the cross-product of the matrices D and S may be calculated as a correlation value. If the images are in binary or grayscale format, then the matrices D and S typically have only a single value each element, so the cross-product may be calculated in a conventional manner. If the images are in color, then the matrices D and S typically have an array of values in each entry, so care must be taken to calculate the cross product using corresponding values from the arrays. Routines for computing cross-products are well known in the art.

[0037] Referring again to Fig. 3, at operation 320 it is determined whether the correlation value determined in operation 315 exceeds a threshold, and if so, then a signal is generated indicating that the threshold was exceeded (operation 325). By contrast, if the correlation value does not exceed a threshold, then a signal is generated indicating that the threshold was not exceeded (operation 330).

[0038] Operation 320 may be implemented by comparing the correlation value determined in operation 315 with a threshold value. The nature of the comparison and the threshold value may differ as a function of the correlation value. By way of example, if the correlation value determined in operation 315 measures the variance (or standard deviation) of the differences between the original digital image and the scanned image, then the variance may be compared to an absolute threshold or to a threshold based on the differences. If the variance (or standard deviation) exceeds the threshold, then control passes to operation 325.

[0039] If the correlation value determined in operation 315 is a cross-product of the matrices D and S, then the threshold may be fixed, or may be a set to a percentage of a maximum correlation. In one implementation, the maximum correlation may be determined by taking the cross-product of the matrix D with itself. If the cross-product of the matrix D with the matrix S exceeds a threshold percentage of the cross-product of D with itself, then control passes to operation 325. The threshold percentage may be a fixed percentage, or may be a variable parameter. Increasing the threshold percentage results in a more stringent test for distinguishing between a printed copy of an original digital image and printed copy of a scanned copy of the original digital image. The threshold constitutes a tradeoff between probability of false positives and probability of miss. Here, the probability of false positive represents the probability of declaring the scanned image as authentic even though this was not the case (*i.e.*, the scanned image was a

forgery). Conversely, probability of miss represents the probability of declaring the scanned image as a forgery even though this was not the case (*i.e.*, the scanned image was indeed authentic).

[0040] The signals generated at operations 325 and 330 may be written to a memory log, presented on a suitable user interface, and/or input to another process executing on a computing device. For example, in a counterfeit detection application, a signal indicating that a threshold has been exceeded may cause a computing device 130 to display a visual and/or audible indication that the scanned image may be a counterfeit, and may cause the computing device to invoke further counterfeit detection operations.

[0041] The printing process and the scanning process may introduce discrepancies between the original digital image and a scanned image of a printed copy of the original digital image. Optionally, a calibration module 208 may implement a process to produce a calibration correlation value (or a set of values) that represents a correlation value between the original digital image and a scanned image of a printed copy of the original digital image. The calibration correlation value may be used in the correlation process.

[0042] **Fig. 7** is a flowchart illustrating exemplary operations for calculating a calibration correlation value. At operation 710 a copy of the original digital image is printed. At operation 715 the printed copy is scanned. At operation 720 a calibration correlation value between the scanned image and the original digital

image is computed. The calibration correlation value may be computed using the techniques described above. At operation 725 the calibration correlation value is stored in memory, *e.g.*, as other data 218. In one alternate embodiment, operation 715-720 may be repeated multiple times and an average calibration correlation value may be stored in memory. In another alternate embodiment, operations 715-720 may be repeated using different printer and scanner combinations, and stored in a memory location that associates a particular printer model and scanner model with a calibration correlation value.

[0043] The calibration correlation value may be used by the correlation module 206 during the correlation process. In one implementation, the calibration correlation value is subtracted from the correlation value computed in operation 315 before a threshold is applied at operation 320.

[0044] Although the described arrangements and procedures have been described in language specific to structural features and/or methodological operations, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or operations described. Rather, the specific features and operations are disclosed as preferred forms of implementing the claimed present subject matter.